

EXHIBIT 1 TO
RESPONSE OF DEC. 1 2002
09/385, 394

PowerPCTM 601

RISC Microprocessor User's Manual



07	06	05	04	11	12	13	14
				03	02	01	00
21	22	23	24	25	26	27	28
0F	0E	0D	0C	0B	0A	09	08
'D'	'C'	'B'	'A'	31	32	33	34
17	16	15	14	13	12	11	10
1F	1E	51	52	1B	'G'	'F'	'E'
		1D	1C		1A	19	18
				61	62	63	64
				23	22	21	20

Figure 2-33. Little-Endian Mapping of Structure S

2.4.5 PowerPC Byte Ordering

The default mapping for PowerPC processors is big-endian. Little-endian mode can be selected after a hard reset by setting the LM bit in the HID0 register in the PPC601 through the use of the `mtspr` instruction in the hard reset handler. The location of the bit is unique for each PowerPC processor.

2.4.6 PowerPC Data Memory with LM Set

One might expect that with the LM bit set (little-endian mode), that the system would have to perform two-, four-, or eight-way byte swaps when transferring a half word, word, or double word between memory and a register. However, the PowerPC architecture emulates little-endian byte ordering by manipulating the three low-order bits of the effective address. No bytes are swapped and individual multiple-byte scalars appear in memory in big-endian order. Setting LM adjusts the way effective addresses are computed without affecting the transfer of data between memory and registers, which is unencumbered by the need for multiplexers to swap bytes.

2.4.6.1 Aligned Scalars

For the load and store instructions listed in Table 2-28, the effective address is computed as specified in the instruction descriptions in Chapter 3, "Addressing Modes and Instruction Set Summary," and is modified as shown in Table 2-29.

Table 2-28. Load/Store Instructions for Data Aligned on Natural Boundaries

Mnemonic	Instruction
<code>lhz</code>	Load Byte and Zero
<code>lbzu</code>	Load Byte and Zero with Update
<code>lbzux</code>	Load Byte and Zero with Update Indexed
<code>lbzx</code>	Load Byte and Zero Indexed
<code>lfd</code>	Load Floating-Point Double-Precision
<code>lfdl</code>	Load Floating-Point Double-Precision with Update